Improved Analysis of a Max Cut Algorithm Based on Spectral Partitioning.

José Soto

October 5, 2009

Abstract

Luca Trevisan [1] presented an approximation algorithm for Max Cut based on spectral partitioning techniques. He proved that the algorithm has an approximation ratio of at least 0.531. We improve this bound up to 0.6142. We also define and extend this result for the more general Maximum Colored Cut problem.

1 Introduction

Given a graph G and a nonnegative weight function on the edges of G, the maximum cut problem is that of finding a partition of the vertices in two sets such that the total weight of the edges crossing the cut is maximized. It is easy to find a solution such that the weight of the cut is at least half of the optimum (for instance, a random partition cuts half of the total weight of the graph). The algorithm of Goemans and Williamson [2], based on semidefinite programming, yields a $0.87856\cdots$ approximation for this problem. Until recently, the only known methods giving a guarantee of more than a half required solving a semidefinite program. In [1], Trevisan presented an algorithm based on spectral partitioning with approximation factor bounded away from half. In that paper, it was shown that the algorithm has an approximation ratio of at least 0.531. We present a new analysis that improves that bound up to 0.6142. We also show how to apply this algorithm to the slightly more general maximum colored cut problem.

2 Algorithm idea

In the Maximum Colored Cut problem (MaxCC), we are given a graph $G = (V, E)$ with $E = R \cup B$, where the edges are colored either red or blue, and a nonnegative weight function $w : E \to \mathbb{R}^+$. We want to partition V into two sets V_- and V_+ such that the weight of the red edges that are cut by the partition plus the weight of the blue edges that are uncut is maximized. This problem generalizes the Max Cut problem (in which $B = \emptyset$).

Given a bipartition of V, we say that a red edge is *good*, if it belongs to the associated cut, and that a blue edge is *good* if it is not cut. Any edge which is not good will be denoted as *bad*. With this notation, the objective of the problem is to find a partition that maximizes the total weight of good edges.

Luca Trevisan's approach [1] to construct an approximation algorithm for Max Cut is the following: Device an algorithm $\mathcal{A}$ that partitions V into three sets V_+, V_0, V_-, where V_0 is the set of nodes that are still "undecided" between V_+ and V_-, and use $\mathcal{A}$ recursively in the set of undecided nodes until every node is decided. For a given tripartition, the edges of the graph induced by $V_+ \cup V_-$ are already labeled as good or bad. Also, no matter what the recursive partition of the undecided vertices into two pieces L and R is, we can always impose that half of the total weight of the edges between $V_+ \cup V_-$ and $V_0 = L \cup R$ is good (by assigning L to V_+ and R to V_- or viceversa). This observation suggests that the objective of $\mathcal{A}$ should be to find a tripartition for which the ratio of the weight of the good edges induced by $V_+ \cup V_-$ plus half of the total weight crossing from $V_+ \cup V_-$ to V_0 with respect to the total weight of the edges involved (the

ones incident in $V_+ \cup V_-$) is as high as possible. We call this ratio the *recoverable ratio* of the partition. As we will see later in the paper, we can find a partition with high recoverable ratio by using spectral partition techniques.

The algorithm we describe is the same as the one in [1], so we assume familiarity with that paper. The previous analysis for the approximation ratio of that algorithm involved upper bounding the number of uncut (bad) edges in each iteration via an application of Cauchy-Schwartz inequality. We give a tighter analysis by directly lower bounding the number of cut (good) edges.

3   Some notation

For a graph $G = (V, E)$ with n vertices, a set of edges $F \subseteq E$, and a set of vertices $A \subseteq V$, we use $F[A]$ to denote the set of edges in F with both endpoints inside of A, similarly, for disjoint $A_1, A_2 \subseteq V$ we use $F(A_1 : A_2)$ to denote the set of edges in F with one endpoint in A_1 and the other in A_2. We define the indicator vector of a bipartition $\{V_+, V_-\}$ of V to be the vector $x \in \{-1, 1\}^n$ such that $x_i = 1$ if $i \in V_+$ and $x_i = -1$ if $i \in V_-$. Similarly we say that $x \in \{-1, 0, 1\}^n$ is the indicator vector of the tripartition $\{V_+, V_0, V_-\}$ of V if x_i is 1, 0 or -1 whenever i is in V_+, V_0 or V_- respectively. As usual, for a weight function $w : E \to \mathbb{R}^+$, and $F \subseteq E$, we use $w(F)$ as a shorthand for $\sum_{e \in E} w(f)$. For the rest of this paper, we fix a graph $G = (V, E)$ where $|V| = n$ and $E = R \cup B$, and a weight function $w : E \to \mathbb{R}^+$.

The following quadratic formulation gives the value of the maximum colored cut.

$$\max_{x \in \{-1,1\}^n} \frac{1}{4} \sum_{\{i,j\} \in R} w_{ij}(x_i - x_j)^2 + \frac{1}{4} \sum_{\{i,j\} \in B} w_{ij}(x_i + x_j)^2. \tag{MaxCC}$$

Let $M^{\{i,j\}} \in \mathbb{R}^{n \times n}$ be the matrix associated to the quadratic form involving edge $\{i, j\}$ for MaxCC:

$$x^T M^{\{i,j\}} x = \begin{cases} w_{ij}(x_i - x_j)^2 & \text{if } \{i,j\} \in R, \\ w_{ij}(x_i + x_j)^2 & \text{if } \{i,j\} \in B. \end{cases}$$

By letting $M = \sum_{\{i,j\} \in E} M_{ij}$, MaxCC can be expressed as $\max_{x \in \{-1,1\}^n} \frac{1}{4} x^T M x$. It will also be convenient to define matrices $D^{\{i,j\}}$ such that

$$x^T D^{\{i,j\}} x = w_{ij}(x_i^2 + x_j^2),$$

and let $D = \sum_{\{i,j\} \in E} D^{\{i,j\}}$. In particular note that if $x \in \{-1, 1\}^n$ then $x^T D x = 2w(E)$. It is also easy to check that

$$M = \text{Deg}(R) - \text{Adj}(R) + \text{Deg}(B) + \text{Adj}(B),$$
$$D = \text{Deg}(R) + \text{Deg}(B).$$

where $\text{Adj}(K)$ and $\text{Deg}(K)$ are the weighted adjacency and degree matrices associated to the edge set K.

In the following lemma, we relate the value of the MaxCC to the highest eigenvalue of a certain positive semidefinite matrix:

Lemma 3.1. *If MaxCC $\geq (1 - \varepsilon)w(E)$, then there exists a vector $x \in \mathbb{R}^n \setminus \{0\}$ such that*

$$\frac{x^T M x}{x^T D x} \geq 2(1 - \varepsilon).$$

Proof. Let x be the vector solving $\max_{x \in \mathbb{R}^n} x^T M x / x^T D x$, and $x^* \in \{-1, 1\}^n$ be the incident vector of a maximum colored cut. Then

$$\frac{x^T M x}{x^T D x} \geq \frac{x^{*T} M x^*}{x^{*T} D x^*} = \frac{4\text{MaxCC}}{2w(E)} \geq 2(1 - \varepsilon).$$

To find x we can simply find a unit vector y associated to the maximum eigenvalue of the positive semidefinite matrix $D^{-1/2} M D^{-1/2}$, and then set $x = D^{-1/2} y$. $\square$

In practice, x can not be found exactly, however, it is know that we can find efficiently a vector x such that $x^T M x \geq 2(1 - \varepsilon - \delta) x^T D x$ in time inverse proportional to δ (see the discussion of Lemma 2 in [1]). In the next section we will show how to derive a good tripartition, this is, one having a good recoverable ratio, using this vector as a starting point.

4 Finding a good tripartition

Let $z \in \{-1, 0, 1\}^n$ be an indicator vector of a partition V_-, V_0, V_+, such that $V \neq V_0$. Define the weights of the good, bad, crossing and incident edges associated to the partition as:

$$\text{Good}(z) := w(R(V_- : V_+)) + w(B[V_-] \cup B[V_+]).$$
$$\text{Bad}(z) := w(B(V_- : V_+)) + w(R[V_-] \cup R[V_+]).$$
$$\text{Cross}(z) := w(E(V_- \cup V_+ : V_0)).$$
$$\text{Inc}(z) := w(E) - w(E[V_0]).$$

Note that $\text{Good}(z) + \text{Bad}(z) + \text{Cross}(z) = \text{Inc}(z)$. The *recoverable ratio* of z is defined to be $(\text{Good}(z) + \text{Cross}(z)/2)$ divided by $\text{Inc}(z)$. In particular, if z is the indicator vector of the optimal MaxCC, the recovery ratio is exactly the ratio $\text{MaxCC}/w(E) = z^T M z / 2 z^t M z$.

Given a vector $x \in \mathbb{R}^n$ such that the ratio $x^T M x / x^T D x$ is big, we will be able to find a rounded vector $z \in \{-1, 0, 1\}^n$ with high recoverable ratio. The following lemma, which can be seen as an improvement of Lemma 3 in [1], will be useful for this task.

Lemma 4.1. *Let $x \in \mathbb{R}^n$, with $||x||_\infty = 1$. Construct $y \in \{-1, 0, 1\}^n$ as follows. Pick $t \in_R [0, 1]$ uniformly at random and let, for each i,*

$$y_i = \begin{cases} 1, & \text{if } x_i \geq \sqrt{t}. \\ -1, & \text{if } x_i \leq -\sqrt{t}. \\ 0, & \text{if } |x_i| \leq \sqrt{t}. \end{cases}$$

Let

$$C(i, j) = \Pr\left[\begin{pmatrix} y_i \\ y_j \end{pmatrix} \in \left\{ \begin{pmatrix} -1 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ -1 \end{pmatrix} \right\} \right],$$

$$U(i, j) = \Pr\left[\begin{pmatrix} y_i \\ y_j \end{pmatrix} \in \left\{ \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \begin{pmatrix} -1 \\ -1 \end{pmatrix} \right\} \right], \text{ and}$$

$$X(i, j) = \Pr\left[\begin{pmatrix} y_i \\ y_j \end{pmatrix} \in \left\{ \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 0 \\ -1 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} -1 \\ 0 \end{pmatrix} \right\} \right],$$

be the probabilities that an edge $\{i, j\}$ is cut, uncut or crossing the tripartition induced by y. Then for all $0 \leq \beta \leq 1$,

$$U(i, j) + \beta X(i, j) \geq \beta(1 - \beta)(x_i + x_j)^2, \tag{4.1}$$
$$C(i, j) + \beta X(i, j) \geq \beta(1 - \beta)(x_i - x_j)^2. \tag{4.2}$$

Proof. Note first that (4.2) follows if we apply (4.1) to the vector x' obtained from x by switching the sign of x_j. To prove (4.1) we consider two cases.

Case 1: If $x_i \cdot x_j \geq 0$. Assume, w.l.o.g. that $|x_i| \leq |x_j|$. In this case, $U(i, j)$ is equal to the probability that both x_i^2 and x_j^2 are bigger than t, thus $U(i, j) = x_i^2$. On the other hand, $X(i, j)$ is equal to the probability that t is between x_i^2 and x_j^2, this is $X(i, j) = (x_j^2 - x_i^2)$.

Using the inequality $\beta(1 - \beta)(a + b)^2 \leq (1 - \beta)a^2 + \beta b^2$, valid for $a, b \geq 0$ and $0 \leq \beta \leq 1$, we get

$$\beta(1 - \beta)(x_i + x_j)^2 = \beta(1 - \beta)(|x_i| + |x_j|)^2 \leq (1 - \beta)x_i^2 + \beta x_j^2 = U(i, j) + \beta X(i, j).$$

3

Case 2: If $x_i \cdot x_j < 0$. Assume again, w.l.o.g. that $|x_i| \leq |x_j|$. In this case it is easy to see that $U(i,j) = 0$ and that $X(i,j) = (x_j^2 - x_i^2)$.

Since $x_i \cdot x_j < 0$, we have $|x_j + x_i| \leq |x_j - x_i|$, and so, using that $0 \leq \beta \leq 1$ we get

$$\beta(1-\beta)(x_i + x_j)^2 \leq \beta|x_i + x_j| \cdot |x_i - x_j| = \beta(x_j^2 - x_i^2) = U(i,j) + \beta X(i,j).$$

$\square$

The main result of this paper is given by the following lemma.

Lemma 4.2. *Given a non-zero vector x such that $x^T M x \geq 2(1-\varepsilon) \cdot x^T D x$, for some $0 \leq \varepsilon \leq 1/2$, we can efficiently find a vector $z \in \{-1, 1, 0\}^n \setminus \{0\}$ such that*

$$\mathrm{Good}(z) + \frac{\mathrm{Cross}(z)}{2} \geq \begin{cases} \dfrac{-1 + \sqrt{4\varepsilon^2 - 8\varepsilon + 5}}{2(1-\varepsilon)} \cdot \mathrm{Inc}(z) & \text{if } \varepsilon \geq \varepsilon_0, \\[3mm] \dfrac{1}{1 + 2\sqrt{\varepsilon(1-\varepsilon)}} \cdot \mathrm{Inc}(z) & \text{if } \varepsilon \leq \varepsilon_0. \end{cases} \tag{4.3}$$

where $\varepsilon_0 \approx 0.22815\ldots$ is the unique solution of the equation $\dfrac{1}{1 + 2\sqrt{\varepsilon(1-\varepsilon)}} = \dfrac{-1 + \sqrt{4\varepsilon^2 - 8\varepsilon + 5}}{2(1-\varepsilon)}$.

Proof. Without loss of generality, assume that $||x||_\infty = 1$. Let y be the random vector obtained from x as in Lemma 4.1, and $0 \leq \beta \leq 1$ to be specified later. Then, using the previous lemma and that $\mathbb{E}[2\mathrm{Inc}(y) - \mathrm{Cross}(y)] = x^T D x$, we obtain

$$\mathbb{E}[\mathrm{Good}(y) + \beta\mathrm{Cross}(y)] = \sum_{\{i,j\} \in R} w_{i,j}(C(i,j) + \beta X(i,j)) + \sum_{\{i,j\} \in B} w_{i,j}(U(i,j) + \beta X(i,j))$$

$$\geq \beta(1-\beta) \sum_{\{i,j\} \in R} w_{i,j}(x_i - x_j)^2 + \beta(1-\beta) \sum_{\{i,j\} \in B} w_{i,j}(x_i + x_j)^2$$

$$= \beta(1-\beta)x^T M x \geq 2(1-\varepsilon)\beta(1-\beta)x^T D x$$

$$= 2(1-\varepsilon)\beta(1-\beta)\mathbb{E}[2\mathrm{Inc}(y) - \mathrm{Cross}(y)].$$

Rearranging terms, we get

$$\mathbb{E}[\mathrm{Good}(y)] + (\beta + 2(1-\varepsilon)\beta(1-\beta))\,\mathbb{E}[\mathrm{Cross}(y)] \geq 4(1-\varepsilon)\beta(1-\beta)\mathbb{E}[\mathrm{Inc}(y)]. \tag{4.4}$$

Let $0 \leq \beta_0 := \dfrac{3 - 2\varepsilon - \sqrt{4\varepsilon^2 - 8\varepsilon + 5}}{4(1-\varepsilon)} \leq \dfrac{1}{2}$ so that $\beta_0 + 2(1-\varepsilon)\beta_0(1-\beta_0) = 1/2$. We have two ways to proceed now. The first one is to set $\beta = \beta_0$ and replace it in (4.4) to obtain:

$$\mathbb{E}[\mathrm{Good}(y)] + \frac{1}{2}\mathbb{E}[\mathrm{Cross}(y)] \geq \frac{-1 + \sqrt{4\varepsilon^2 - 8\varepsilon + 5}}{2(1-\varepsilon)}\mathbb{E}[\mathrm{Inc}(y)]. \tag{4.5}$$

Observe that this bound is valid for all $0 \leq \varepsilon \leq 1/2$ but it gets worse as ε gets closer to 0 (for every value of ε, the optimal recovery ratio is at least $(1-\varepsilon)$).

The second way to select β will give us a bound that is tighter when ε is close to 0. Recall that $\mathrm{Good}(y) + \mathrm{Bad}(y) + \mathrm{Cross}(y) = \mathrm{Inc}(y)$. Subtracting (4.4) from this we get

$$(1 - 4(1-\varepsilon)\beta(1-\beta))\mathbb{E}[\mathrm{Inc}(y)] \geq \mathbb{E}[\mathrm{Bad}(y)] + \big(1 - \beta - 2(1-\varepsilon)\beta(1-\beta)\big)\mathbb{E}[\mathrm{Cross}(y)]. \tag{4.6}$$

Observe that the coefficient in front of $\mathbb{E}[\mathrm{Cross}(y)]$, $1 - \beta - 2(1-\varepsilon)\beta(1-\beta) = (1-\beta)(1 - 2(1-\varepsilon)\beta)$ is positive and at most $1/2$ whenever $\beta_0 \leq \beta < 1/(2-2\varepsilon)$. Assume that β is in that interval and divide the equation (4.6) by $2(1-\beta)(1 - 2(1-\varepsilon)\beta)$ to obtain

$$\frac{1 - 4(1-\varepsilon)\beta(1-\beta)}{2(1-\beta)(1 - 2(1-\varepsilon)\beta)}\mathbb{E}[\mathrm{Inc}(y)] \geq \frac{\mathbb{E}[\mathrm{Bad}(y)]}{2(1-\beta)(1 - 2(1-\varepsilon)\beta)} + \frac{1}{2}\mathbb{E}[\mathrm{Cross}(y)] \geq \mathbb{E}[\mathrm{Bad}(y)] + \frac{1}{2}\mathbb{E}[\mathrm{Cross}(y)].$$

4

Since $\text{Good}(y) + \text{Bad}(y) + \text{Cross}(y) = \text{Inc}(y)$, we get

$$\mathbb{E}[\text{Good}(y)] + \frac{1}{2}\mathbb{E}[\text{Cross}(y)] \geq \left(1 - \frac{1 - 4(1-\varepsilon)\beta(1-\beta)}{2(1-\beta)(1-2(1-\varepsilon)\beta)}\right)\mathbb{E}[\text{Inc}(y)] = \frac{(1-2\beta)}{2(1-\beta)(1-2(1-\varepsilon)\beta)}\mathbb{E}[\text{Inc}(y)].$$

The function $F(\beta) := \dfrac{(1-2\beta)}{2(1-\beta)(1-2(1-\varepsilon)\beta)}$ is nonnegative whenever $\beta \leq 1/2$ or $1/(2-2\varepsilon) < \beta < 1$. Because of our previous assumption on β, the only case of interest is when $\beta_0 \leq \beta \leq 1/2$.

A little computation shows that $F(\beta)$ has a local maximum in $\beta^* = \dfrac{1}{2} - \dfrac{1}{2}\sqrt{\dfrac{\varepsilon}{1-\varepsilon}} \leq \dfrac{1}{2}$, with value

$F(\beta^*) = \dfrac{1}{1 + 2\sqrt{\varepsilon(1-\varepsilon)}}$. Hence, provided that $\beta_0 \leq \beta^*$, we can set $\beta = \beta^*$ to obtain

$$\mathbb{E}[\text{Good}(y)] + \frac{1}{2}\mathbb{E}[\text{Cross}(y)] \geq \frac{1}{1 + 2\sqrt{\varepsilon(1-\varepsilon)}}\mathbb{E}[\text{Inc}(y)]. \tag{4.7}$$

Let $\varepsilon_0 \approx 0.22815\ldots$ be the unique solution of the equation $\dfrac{1}{1 + 2\sqrt{\varepsilon(1-\varepsilon)}} = \dfrac{-1 + \sqrt{4\varepsilon^2 - 8\varepsilon + 5}}{2(1-\varepsilon)}$. It is easy to check that $\beta_0 \leq \beta^*$ if and only if $\varepsilon \leq \varepsilon_0$. This, together with (4.5) and (4.7), implies that the desired inequality, (4.3), holds in expectation and therefore, there exists a $z \in \{-1, 0, 1\}^n$ satisfying it. In particular, we can find z by testing all n thresholds (x_i^2) for t, and keeping the one with greatest recoverable ratio. Furthermore, this vector z is not zero, since for the index i such that $|x_i| = 1$, we also have $|z_i| = 1$. $\square$

Remark 4.3. Consider the function

$$f(\varepsilon) := \begin{cases} \dfrac{-1 + \sqrt{4\varepsilon^2 - 8\varepsilon + 5}}{2(1-\varepsilon)}, & \text{if } \varepsilon \geq \varepsilon_0, \\[3mm] \dfrac{1}{1 + 2\sqrt{\varepsilon(1-\varepsilon)}}, & \text{if } \varepsilon \leq \varepsilon_0. \end{cases}$$

Lemmas 3.1 and 4.2 imply that if G has a bipartition with at least $1 - \varepsilon$ fraction of the total weight being good ('cut'), then we can find a partition into three pieces V_-, V_+, V_0 with recoverable ratio at least $f(\varepsilon)$. Previously, Luca Trevisan proved (analyzing Min Uncut) that the same is true for the function $\tilde{f}(\varepsilon) = 1 - 2\sqrt{\varepsilon}$. In Figure 1 we can see the relationship between these two bounds. Note that we always have $f(\varepsilon) \geq \tilde{f}(\varepsilon)$. Also, it is worth noting that a random bipartition gives a recoverable ratio of $1/2$ in expectation. The previous guarantee only beats $1/2$ when $\varepsilon < 1/16$, the new guarantee implies that the algorithm is better than a random cut for any $\varepsilon < 1/3$.

5   Main algorithm

For a given $G = (V, E = R \cup B)$ and nonnegative weight function $w : E \to \mathbb{R}^+$, the following algorithm returns a bipartition (W_+, W_-) that is a good approximation to MaxCC problem. (In what follows, we assume that the vector x given by Lemma 3.1 can be found exactly in order to keep the argument simpler).

ALG(V, E)

1. Compute x using Lemma 3.1 and determine from it the vector z given by Lemma 4.2.

2. If $\dfrac{\text{Good}(z) + \text{Cross}(z)/2}{Inc(z)} < 1/2$, then return a bipartition (W_+, W_-) of V such that the weight of good edges is at least half of $w(E)$ (a random cut suffices).

3. Otherwise, let (V_+, V_-, V_0) be the tripartition induced by z. If $V_0 = \emptyset$ return (V_+, V_-). Else, let $(W_+, W_-) \leftarrow$ ALG$(V_0, E[V_0])$, and return the best of $(V_+ \cup W_+, V_- \cup W_-)$ and $(V_+ \cup W_-, V_- \cup W_+)$.
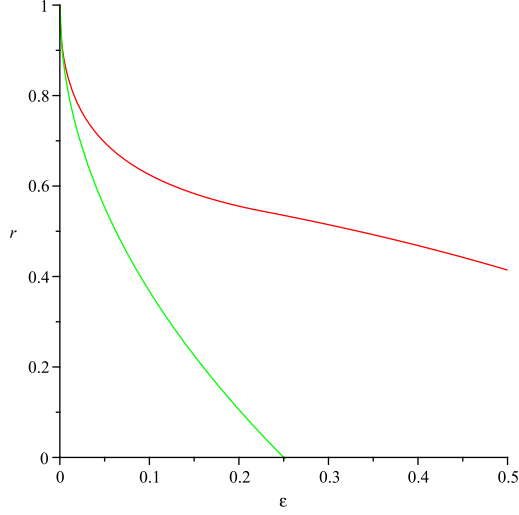
Figure 1: The red curve (top) corresponds to the recoverable ratio guaranteed by $f(\varepsilon)$, the green one (lowest) to the one guaranteed by $\tilde{f}(\varepsilon)$.

Theorem 5.1. *If $MaxCC(G) = (1 - \varepsilon)w(E)$, then $\mathtt{ALG}(V, E)$ returns a partition (W_-, W_+), such that, if $y \in \{-1, 1\}^n$ is the associated indicator vector,*

$$\frac{\text{Good}(y)}{w(E)} \geq \int_0^1 \max\left(\frac{1}{2}, f(\varepsilon/r)\right) dr$$

with $f(\varepsilon)$ as in remark 4.3.

Proof. First, it is clear that the algorithm terminates since in every recursive call the residual graph considered has at least one fewer node. Assume that the algorithm performs T recursive calls. Let $G_t = (V_t, E_t)$ be the graph at the beginning of the t-th iteration, so that $G_1 = (V, E)$. Let also G_{T+1} to be the empty graph and $\delta_t = w(E_t)/w(E)$ for every t. We observe that

$$\frac{w(E_t) - \text{MaxCC}(G_t)}{w(E_t)} \leq \frac{w(E) - \text{MaxCC}(G)}{\delta_t \cdot w(E)} \leq \frac{\varepsilon}{\delta_t}.$$

The previous holds since for the optimal partition defining $\text{MaxCC}(G)$, the total weight of the bad edges is at least the weight of the bad edges inside the subgraph induced by E_t. This quantity is itself at least the weight of the bad edges for the partition defining $\text{MaxCC}(G_t)$. From this observation we get that $\text{MaxCC}(G_t) \geq (1 - \varepsilon/\delta_t)w(E_t)$. Recall that a random cut has a recoverable ratio of at least half. Combining this with Remark 4.3 we get that inside $E_t \setminus E_{t+1}$ we are guaranteed a good weight of at least

$$\max\left(\frac{1}{2}, f(\varepsilon/\delta_t)\right) \cdot w(E_t \setminus E_{t+1}).$$

Using that f is decreasing, and that the sets $E_t \setminus E_{t+1}$ are mutually disjoint, we obtain that the total good

weight returned by the algorithm is at least

$$\frac{\text{Good}(y)}{w(E)} \geq \sum_{i=0}^{T} \max\left(\frac{1}{2}, f(\varepsilon/\delta_t)\right) \cdot (\delta_t - \delta_{t+1})$$

$$= \sum_{i=0}^{T} \int_{\delta_{t+1}}^{\delta_t} \max\left(\frac{1}{2}, f(\varepsilon/\delta_t)\right) dr$$

$$\geq \sum_{i=0}^{T} \int_{\delta_{t+1}}^{\delta_t} \max\left(\frac{1}{2}, f(\varepsilon/r)\right) dr$$

$$= \int_0^1 \max\left(\frac{1}{2}, f(\varepsilon/r)\right) dr.$$

$\square$

Corollary 5.2. *The partition returned by* `ALG` *guarantees a 0.6142... approximation for MaxCC.*

Proof. Let $F(\varepsilon) = \int_0^1 \max\left(1/2, f(\varepsilon/r)\right) dr$. By Theorem 5.1, $F(\varepsilon)$ is the fraction of the weight guaranteed to be good ('cut') by the algorithm. Note that $f(x) = 1/2$ when $x = 1/3$. Using this, and the definition of f it is easy to see that

$$F(\varepsilon) = \begin{cases} 1/2, & \text{if } \varepsilon \geq 1/3. \\[2mm] \displaystyle\int_0^{3\varepsilon} \frac{1}{2} dr + \int_{3\varepsilon}^1 \frac{-1 + \sqrt{4(\varepsilon/r)^2 - 8(\varepsilon/r) + 5}}{2(1 - \varepsilon/r)} dr, & \text{if } \varepsilon_0 \leq \varepsilon \leq 1/3. \\[2mm] \displaystyle\int_0^{3\varepsilon} \frac{1}{2} dr + \int_{3\varepsilon}^{\varepsilon/\varepsilon_0} \frac{-1 + \sqrt{4(\varepsilon/r)^2 - 8(\varepsilon/r) + 5}}{2(1 - \varepsilon/r)} dr + \int_{\varepsilon/\varepsilon_0}^1 \frac{1}{1 + 2\sqrt{\varepsilon/r(1 - \varepsilon/r)}} dr, & \text{if } \varepsilon \leq \varepsilon_0. \end{cases}$$

with $\varepsilon_0 \approx 0.228155\ldots$, as in Lemma 4.2.

The guarantee of approximation of the algorithm is, then, the minimum of the function $G(\varepsilon) = \frac{F(\varepsilon)}{1-\varepsilon}$ when ε is in the interval $[0, 1/2]$. We can check that the function $G(\varepsilon)$ is convex and has an unique minimum at $\varepsilon^* \approx .11089\ldots$ with value $G(\varepsilon^*) \approx 0.614247\ldots$ $\square$

For completeness, we include a closed form of function G and a plot comparing this guarantee with the previous guarantee of [1] for different regimes of ε.

5.1 Analytic expression of the guarantee function.

If $\text{MaxCC}(G) \geq w(E) \cdot (1 - \varepsilon)$, then the algorithm described before gives a $G(\varepsilon)$-approximation for the MaxCC problem, with $G(\varepsilon)$ defined by the following expression:

If $\varepsilon \geq 1/3$

$$G(\varepsilon) := \frac{1}{2(1 - \varepsilon)}.$$

If $\varepsilon_0 \leq \varepsilon \leq 1/3$,

$$G(\varepsilon) := \frac{1}{2(1-\varepsilon)} \cdot \left(\varepsilon - 1 + \sqrt{4\varepsilon^2 - 8\varepsilon + 5} - \varepsilon \ln\left(\frac{1 + \sqrt{4\varepsilon^2 - 8\varepsilon + 5}}{8\varepsilon} \right) \right.$$

$$\left. + \frac{\sqrt{5}}{5} \varepsilon \ln\left(\frac{5 - 4\varepsilon + \sqrt{5(4\varepsilon^2 - 8\varepsilon + 5)}}{(11 + 5\sqrt{5})\,\varepsilon} \right) \right),$$

7

and if $\varepsilon \le \varepsilon_0$,

$$G(\varepsilon) := \frac{1}{2(1-\varepsilon)} \cdot \left(\varepsilon\left(1 - \frac{3}{\varepsilon_0}\right) + 2 + \frac{\varepsilon}{\varepsilon_0}\sqrt{4\varepsilon_0^2 - 8\varepsilon_0 + 5} - \varepsilon\ln\left(\frac{1 + \sqrt{4\varepsilon_0^2 - 8\varepsilon_0 + 5}}{8\varepsilon_0}\right) \right.$$

$$+ \frac{\sqrt{5}}{5}\varepsilon\ln\left(\frac{5 - 4\varepsilon_0 + \sqrt{5(4\varepsilon_0^2 - 8\varepsilon_0 + 5)}}{(11 + 5\sqrt{5})\varepsilon_0}\right) + 16\varepsilon\ln\left(\frac{\sqrt{\varepsilon} + \sqrt{1-\varepsilon}}{\sqrt{\varepsilon} + \sqrt{\frac{\varepsilon}{\varepsilon_0} - \varepsilon}}\right)$$

$$\left. + 8\varepsilon\frac{\sqrt{\varepsilon_0(1-\varepsilon_0)} + 1 - 2\varepsilon_0}{\varepsilon_0 + \sqrt{\varepsilon_0(1-\varepsilon_0)}} - 8\sqrt{\varepsilon}\frac{\sqrt{\varepsilon(1-\varepsilon)} + 1 - 2\varepsilon)}{\sqrt{\varepsilon} + \sqrt{1-\varepsilon}} \right).$$

Where $\varepsilon_0 \approx 0.22815\ldots$ is the unique solution of the equation $\dfrac{1}{1 + 2\sqrt{\varepsilon(1-\varepsilon)}} = \dfrac{-1 + \sqrt{4\varepsilon^2 - 8\varepsilon + 5}}{2(1-\varepsilon)}$.
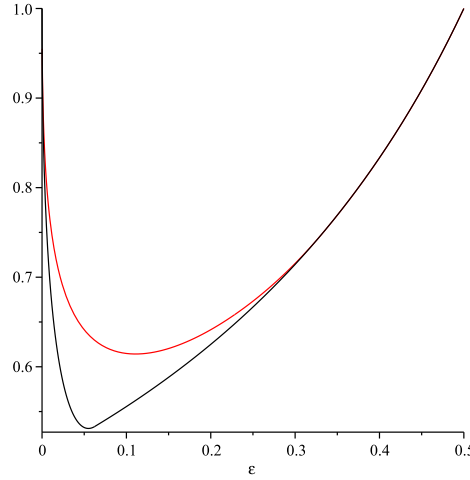

Figure 2: The red curve corresponds to $G(\varepsilon) = \frac{F(\varepsilon)}{1-\varepsilon}$, the black one bellow correspond to the guarantee $H(\varepsilon)$ given previously in [1] with minimum on 0.531: $H(\varepsilon) = \frac{1 - 4\sqrt{\varepsilon} + 8\varepsilon}{1-\varepsilon}$, for $\varepsilon \le 1/16$ and $H(\varepsilon) = \frac{1}{2(1-\varepsilon)}$, for $\varepsilon \ge 1/16$.

References

[1] L. Trevisan. Max Cut and the Smallest Eigenvalue, *In Proc. of 41st ACM STOC, 2009.*

[2] M.X. Goemans and D.P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming, *Journal of the ACM (JACM), 42(6):1115-1145, 1995*